

Основные обозначения

$ A $	– мощность множества A ;
\tilde{a}	– статистическая оценка параметра a распределения случайной величины;
c_i	– центр класса ω_i (кластера X_i);
$d(\mathbf{x})$	– решающая (дискриминантная) функция;
$d(\mathbf{x}, \mathbf{y})$	– метрика (функция расстояния) между векторами \mathbf{x} и \mathbf{y} ;
$d_p(\mathbf{x}, \mathbf{y}) = \ \mathbf{x} - \mathbf{y}\ _p$	– метрика Минковского ($1 \leq p \leq \infty$);
$d_2(\mathbf{x}, \mathbf{y}) = \ \mathbf{x} - \mathbf{y}\ _2$	– метрика Евклида;
$d_{S^{-1}}(\mathbf{x}, \mathbf{y}) = \ \mathbf{x} - \mathbf{y}\ _{S^{-1}}$	– метрика Махалобиса;
$d_k(\mathbf{x}, \mathbf{y})$	– метрика Канберра;
$d(\mathbf{x}, \varpi)$	– расстояние между вектором \mathbf{x} и классом ϖ ;
$d(\omega_i, \varpi_j)$	– расстояние между классами ω_i и ω_j ;
I	– единичная матрица;
$F(\mathbf{w})$	– функция критерия (функционал ошибки);
$M[\cdot]$	– оператор математического ожидания;
m	– количество классов;
N	– количество элементов обучающего множества;
n	– размерность пространства признаков R^n ;
(r_{ij})	– платежная матрица;
$\text{sgn}(t)$	– функция знака (сигнум);
$u(\mathbf{x}, \mathbf{y})$	– потенциальная функция;
\mathbf{v}^T	– транспонирование вектора \mathbf{v} ;
$\mathbf{w} = (w_i)$	– вектор весов дискриминантной функции;
X_i	– область предпочтения класса ϖ_i ;
x	– образ;
$\mathbf{x} = (x_1, x_2, \dots)^T$	– вектор-столбец признаков образа x ;
x_i	– i -й признак образа x ;
(\mathbf{x}_i, y_i)	– i -й прецедент;
$Y = \{y_1, \dots, y_m\}$	– множество меток классов;
$\{\varphi_j(\mathbf{x})\}$	– полная ортогональная система функций;
$\eta(t)$	– функция Хэвисайда;
$\Omega = \{\varpi_1, \dots, \varpi_m\}$	– множество классов;
ϖ_i	– i -й класс;
$\Xi = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$	– обучающее множество (выборка);
(Ξ, Y)	– множество прецедентов;

4. Алгоритмы кластеризации (векторного квантования)

4.1. Постановка задачи кластеризации

Идея векторного квантования состоит в разбиении обучающей выборки $\Xi = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ на непересекающиеся подмножества-кластеры X_1, \dots, X_m : $X_1 \cup \dots \cup X_m = \Xi$, $X_i \cap X_j = \emptyset$ для всех $i \neq j$, таким образом, чтобы все точки одного кластера состояли из «похожих» элементов, а точки разных кластеров существенно отличались. Эта задача является очень неопределенной, так как ее решение зависит от нескольких факторов – параметров кластеризации: 1) выбранного критерия «похожести» элементов Q ; 2) от используемой метрики d (критерии похожести, как правило, зависят от выбранной метрики, измеряющей расстояние между векторами-образами); 3) от установленного (или оцениваемого) числа кластеров.

Выбор параметров кластеризации, как правило, неоднозначен, зачастую субъективен, но этот выбор должен быть согласован с целями кластеризации. Среди основных целей кластеризации могут быть следующие:

1) кластеризация проводится для нахождения групп схожих элементов с целью дальнейшей независимой их обработки. В этом случае параметры кластеризации должны обеспечивать минимальность числа кластеров;

2) кластеризация осуществляется с целью получения новой небольшой выборки, состоящей из эталонных элементов – типичных представителей кластеров. Здесь важно, чтобы параметры кластеризации обеспечивали формирование кластеров с высокой степенью однородности входящих в них элементов;

3) кластеризация проводится с целью нахождения нетипичных элементов, т.е. элементов, не попадающих ни в один из кластеров, при этом сами кластеры должны быть небольшими;

4) кластеризация осуществляется с целью формирования иерархической структуры выборки (так называемая *задача таксономии*). В этом случае на каждом иерархическом уровне количество кластеров должно быть небольшим.

В общем случае задачу кластеризации можно сформулировать следующим образом. Дана обучающая выборка $\Xi = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$. Требуется найти такую функцию кластеризации f , которая каждой точке $\mathbf{x} \in \Xi$ ставила бы в однозначное соответствие некоторый элемент – метку $y \in Y$ из множества меток $Y = \{y_1, \dots, y_m\}$ (каждая метка y_i соответствует некоторому кластеру X_i). В задаче кластеризации множество меток Y неизвестно (и даже неиз-

вестна мощность этого множества: $m = |Y| \ll N$). Если множество Y известно, то задача кластеризации вырождается в задачу классификации.

Множество меток Y необходимо искать среди множества Ψ допустимых меток для данной задачи кластеризации, которое определяется целями кластеризации. Тогда задача кластеризации сводится к нахождению такого множества меток Y_0 и функции кластеризации f , чтобы

$$Y_0 = \arg \min_{Y \in \Psi, f} Q(Y, f),$$

где $Q(Y, f)$ – выбранный критерий качества (оптимальности) кластеризации (если оптимальность соответствует минимуму критерия Q).

Среди критериев оптимальности (качества) кластеризации выделяют следующие:

- 1) среднее внутрикластерное расстояние $Q^{(1)} = \sum_i \sum_{\mathbf{x}, \mathbf{y} \in X_i} d(\mathbf{x}, \mathbf{y}) \rightarrow \min$;
- 2) среднее межкластерное расстояние $Q^{(2)} = \sum_{i < j} \sum_{\substack{\mathbf{x} \in X_i, \\ \mathbf{y} \in X_j}} d(\mathbf{x}, \mathbf{y}) \rightarrow \max$;
- 3) суммарная выборочная дисперсия разброса элементов относительно центров кластеров $Q^{(3)} = \sum_i \frac{1}{|X_i|} \sum_{\mathbf{x} \in X_i} d^2(\mathbf{x}, \mathbf{c}_i) \rightarrow \min$, где $\mathbf{c}_i = \frac{1}{|X_i|} \sum_{\mathbf{x} \in X_i} \mathbf{x}$ – центр кластера X_i .

Заметим, что для использования последнего критерия оптимальности необходимо, чтобы пространство признаков было не только метрическим, но и линейным (т.е. в этом пространстве можно было осуществлять сложение и умножение на число векторов-признаков). Если пространство не является линейным, то вычислительная сложность алгоритмов кластеризации значительно увеличивается. Действительно, для вычисления центра \mathbf{c} кластера X в линейном пространстве требуется $O(|X|)$ операций, а в метрическом пространстве (здесь в качестве центра можно взять точку $\mathbf{c} = \arg \min_{\mathbf{x} \in X} \sum_{\mathbf{y} \in X} d(\mathbf{x}, \mathbf{y})$) – $O(|X|^2)$.

На практике вместо одного критерия оптимальности используется несколько критериев. Например, критерий $Q^{(1)}/Q^{(2)} \rightarrow \min$, учитывающий как межкластерные, так и внутрикластерные расстояния.

4.2. Алгоритм k -внутригрупповых средних (k -means)

Рассмотрим один из популярных алгоритмов кластеризации, основанный на минимизации функционала суммарной выборочной дисперсии разброса элементов относительно центров тяжести кластеров $Q = Q^{(3)}$. Этот алгоритм представляет собой пошаговое (итерационное) нахождение центров тяжести кластеров и разбиение обучающей выборки на кластеры до тех пор, пока функционал Q не перестанет уменьшаться.

Алгоритм k -means

1. Выделяются некоторые образы из обучающей выборки – начальные цен-

тры кластеров $\mathbf{c}_1^{(0)}, \dots, \mathbf{c}_m^{(0)}$ и полагается $k = 0$.

2. Вся обучающая выборка разбивается на m кластеров (клеток Вороного) по методу ближайшего соседа – получаются некоторые кластеры $X_1^{(k)}, \dots, X_m^{(k)}$.

3. Рассчитываются новые центры – центры тяжести кластеров по формуле $\mathbf{c}_i^{(k+1)} = \frac{1}{|X_i^{(k)}|} \sum_{\mathbf{x} \in X_i^{(k)}} \mathbf{x}$.

4. Проверяется выполнение условия останова: $\mathbf{c}_i^{(k+1)} = \mathbf{c}_i^{(k)}$ для всех $k = 1, \dots, m$. В противном случае – переход к пункту 2.

Теорема 4.1. Алгоритм *k-means* минимизирует функционал суммарной выборочной дисперсии $Q^{(3)}$ и сходится за конечное число шагов.

Доказательство. Покажем, что в процессе выполнения шагов алгоритма минимизируется функционал $Q = Q^{(3)}$. Действительно, сначала (пункт 2 алгоритма) он минимизируется при фиксированном положении центров тяжести кластеров путем оптимизации разбиения обучающей выборки на кластеры

$$\mathbf{x} \in X_i, \text{ если } \|\mathbf{x} - \mathbf{c}_i^{(l)}\| \leq \|\mathbf{x} - \mathbf{c}_j^{(l)}\| \text{ для всех } j \neq i.$$

Покажем, что в пункте 3 алгоритма осуществляется минимизация функционала Q за счет пересчета центров тяжести кластеров при фиксированном разбиении обучающей выборки на кластеры

$$\mathbf{c}_i^{(l+1)} = \frac{1}{|X_i^{(l)}|} \cdot \sum_{\mathbf{x} \in X_i^{(l)}} \mathbf{x}.$$

Для этого рассмотрим функцию разброса $R(\mathbf{c}_i)$ выборочных значений в i -м классе относительно некоторой точки \mathbf{c}_i (не обязательно центра класса):

$R(\mathbf{c}_i) = \sum_{\mathbf{x} \in X_i} \|\mathbf{x} - \mathbf{c}_i\|^2 = \sum_{\mathbf{x} \in X_i} (\mathbf{x}^2 - 2\mathbf{x} \cdot \mathbf{c}_i + \mathbf{c}_i^2)$. Исследуем на минимум эту функцию методом дифференциального исчисления. Имеем

$$\frac{\partial R}{\partial c_{ik}} = \left(\sum_{\mathbf{x} \in X_i} \sum_{k=1}^n (x_k - c_{ik})^2 \right)' = 2 \sum_{\mathbf{x} \in X_i} (x_k - c_{ik}) \cdot (-1) = 0.$$

Откуда $c_{ik} = \frac{1}{|X_i|} \sum_{\mathbf{x} \in X_i} x_k$. Следовательно, минимум функции $R(\mathbf{c}_i)$ достигается при $\mathbf{c}_i = \frac{1}{|X_i|} \sum_{\mathbf{x} \in X_i} \mathbf{x}$.

Таким образом, на каждой итерации значение Q не увеличивается, т.е. $Q_{(1)} \geq Q_{(2)} \geq \dots$. Имеем невозрастающую последовательность $Q_{(l)}$, ограниченную снизу нулем, поэтому эта последовательность имеет предел. Так как число элементов обучающей выборки, а, следовательно, и число различных разбиений, конечно, то этот предел достигается за конечное число итераций. ■

Замечания.

1. Алгоритм k -means осуществляет локальную, но не глобальную минимизацию функционала Q . Поэтому гарантии «хорошей» кластеризации этот алгоритм не дает.

2. Существует много алгоритмов векторного квантования, похожих на k -means, но обучающихся быстрее. Правда качество такого обучения может быть хуже, чем в k -means.

3. Процедура k -means относится к алгоритмам обучения без учителя (с самообучением).

4. Векторное квантование очень чувствительно к размерности пространства признаков: требуемое количество центров кластеров экспоненциально растет с ростом размерности. Поэтому, если удастся избавиться от признака, мало влияющего на классификацию, то векторное квантование начинает работать быстрее и лучше.

5. Рассматриваются и **невекторные** методы квантования. В этих методах осуществляется квантование не образов – отдельных векторов, а орбит – образов относительно некоторой группы преобразований, не влияющих на кластеризацию (например, сдвиги, растяжения, небольшие искажения букв, цифр).

Пример. Предположим, что на плоскости R^2 заданы векторы-образы $\mathbf{x}_1 = (1,1)$, $\mathbf{x}_2 = (0,0)$, $\mathbf{x}_3 = (2,0)$, $\mathbf{x}_4 = (4,4)$, $\mathbf{x}_5 = (5,5)$, $\mathbf{x}_6 = (5,3)$ (рис. 4.1). Найдем кластеризацию этих образов по двум классам. Для этого выполним последовательно шаги рассмотренного алгоритма.

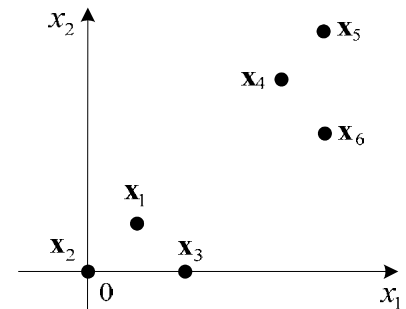


Рис. 4.1

1. В качестве начальных центров кластеров выберем образы $\mathbf{c}_1^{(0)} = \mathbf{x}_1$ и $\mathbf{c}_2^{(0)} = \mathbf{x}_2$. Тогда, разбивая выборку $\{\mathbf{x}_1, \dots, \mathbf{x}_6\}$ на два подмножества по методу ближайшего соседа, получим начальные кластеры $X_1^{(0)} = \{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ и $X_2^{(0)} = \{\mathbf{x}_2\}$.

2. Вычисляем новые центры – центры тяжести кластеров

$$\mathbf{c}_1^{(1)} = \frac{1}{5} \begin{pmatrix} x_{11} + x_{31} + x_{41} + x_{51} + x_{61} \\ x_{12} + x_{32} + x_{42} + x_{52} + x_{62} \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 1 + 2 + 4 + 5 + 5 \\ 1 + 0 + 4 + 5 + 3 \end{pmatrix} = \begin{pmatrix} 17/5 \\ 13/5 \end{pmatrix}, \quad \mathbf{c}_2^{(1)} = \mathbf{x}_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

3. Сравниваем: $\mathbf{c}_1^{(0)} \neq \mathbf{c}_1^{(1)}$ и $\mathbf{c}_2^{(0)} = \mathbf{c}_2^{(1)}$. Продолжаем выполнение алгоритма.

4. Разбиваем выборку $\{\mathbf{x}_1, \dots, \mathbf{x}_6\}$ на два подмножества с новыми центрами по методу ближайшего соседа, получим кластеры $X_1^{(1)} = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ и $X_2^{(1)} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$.

5. Вновь вычисляем центры тяжести кластеров

$$\mathbf{c}_1^{(2)} = \frac{1}{3} \begin{pmatrix} x_{41} + x_{51} + x_{61} \\ x_{42} + x_{52} + x_{62} \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 4 + 5 + 5 \\ 4 + 5 + 3 \end{pmatrix} = \begin{pmatrix} 14/3 \\ 4 \end{pmatrix},$$

$$\mathbf{c}_2^{(2)} = \frac{1}{3} \begin{pmatrix} x_{11} + x_{21} + x_{31} \\ x_{12} + x_{22} + x_{32} \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 + 0 + 2 \\ 1 + 0 + 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1/3 \end{pmatrix}.$$

6. Сравниваем: $\mathbf{c}_1^{(1)} \neq \mathbf{c}_1^{(2)}$ и $\mathbf{c}_2^{(1)} \neq \mathbf{c}_2^{(2)}$. Продолжаем выполнение алгоритма.

7. Разбиваем выборку $\{\mathbf{x}_1, \dots, \mathbf{x}_6\}$ на два подмножества с новыми центрами по методу ближайшего соседа, получим кластеры $X_1^{(2)} = \{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ и $X_2^{(2)} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$.

8. Вновь вычисляем центры тяжести кластеров $\mathbf{c}_1^{(3)} = \mathbf{c}_1^{(2)}$, $\mathbf{c}_2^{(3)} = \mathbf{c}_2^{(2)}$. Остановка алгоритма.

При практической реализации алгоритма возникают следующие проблемы:

1) необходимо задать число кластеров;

2) качество работы алгоритма зависит от начальной расстановки центров кластеров.

Для решения этих проблем рекомендуется осуществить несколько кластеризаций при разных начальных расстановках центров кластеров и различных значений числа кластеров. После чего необходимо выбрать ту кластеризацию, которая доставляет минимум функционалу $Q^{(3)}$.

4.3. Алгоритмы расстановки центров кластеров

Для первоначальной расстановки центров кластеров применяются следующие алгоритмы, которые можно рассматривать и как самостоятельные алгоритмы кластеризации.

4.3.1. Алгоритм простейшей расстановки центров кластеров

Вводится некоторый порог $h > 0$, в качестве первого центра кластера назначается первый элемент выборки $\mathbf{c}_1 = \mathbf{x}_1$.

Предположим, что уже выбраны k центров кластеров. Тогда в качестве очередного $k+1$ -го центра кластера выбирается такой элемент выборки \mathbf{x}_j , что минимальное расстояние от \mathbf{x}_j до центров \mathbf{c}_i , $i = 1, \dots, k$, будет больше h .

4.3.2. Алгоритм, основанный на методе просеивания

В этом алгоритме рассматривается некоторая неотрицательная функция $f(\mathbf{x})$, называемая *плотностью распределения* элементов обучающей выборки и принимающая тем большее значение, чем ближе элемент \mathbf{x} расположен к точке сгущения элементов выборки. Например, в качестве $f(\mathbf{x})$ можно взять следующую функцию:

$$f(\mathbf{x}) = f_h(\mathbf{x}) = \frac{1}{h^2} \cdot \sum_{i: \|\mathbf{x} - \mathbf{x}_i\| < h} (h^2 - \|\mathbf{x} - \mathbf{x}_i\|^2),$$

где $h > 0$ – некоторое пороговое значение. Затем осуществляется упорядочивание элементов обучающей выборки таким образом, чтобы $f(\mathbf{x}_1) \geq f(\mathbf{x}_2) \geq f(\mathbf{x}_3) \geq \dots$. Далее осуществляется алгоритм простейшей расстановки центров кластеров, в котором в первую очередь в качестве новых центров кластеров выбираются те элементы обучающей выборки, в которых значение плотности будет наибольшим.

4.3.3. Алгоритм максиминного расстояния

Этот алгоритм состоит из следующих шагов.

Максиминный алгоритм

1. В качестве первого центра кластера выбирается элемент $\mathbf{c}_1 = \mathbf{x}_1$.
2. В качестве второго центра кластера выбирается тот элемент $\mathbf{c}_2 = \mathbf{x}_{j_2}$, который находится на наибольшем расстоянии от \mathbf{c}_1 , т.е. $\|\mathbf{x}_{j_2} - \mathbf{c}_1\| = \max_{\mathbf{x} \in \Xi} \|\mathbf{x} - \mathbf{c}_1\|$.
3. Предположим, что выбраны k центров $C^{(k)} = \{\mathbf{c}_1, \dots, \mathbf{c}_k\}$ кластеров. В качестве очередного $(k+1)$ -го центра кластера выбирается тот элемент $\mathbf{x}_{j_{k+1}}$, который находится на наибольшем расстоянии от ближайшего из центров $\mathbf{c}_1, \dots, \mathbf{c}_k$ (рис. 4.2), т.е. $\min_{\mathbf{c} \in C^{(k)}} \|\mathbf{x}_{j_{k+1}} - \mathbf{c}\| = \max_{\mathbf{x} \in \Xi \setminus C^{(k)}} \min_{\mathbf{c} \in C^{(k)}} \|\mathbf{x} - \mathbf{c}\|$.
4. Проверяется условие останова.

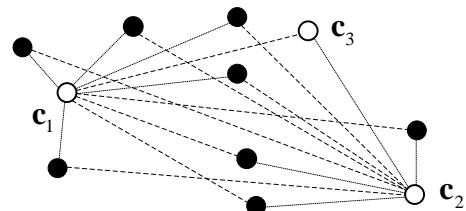


Рис. 4.2

Условием останова алгоритма может быть выполнение неравенства $Q_{(k+1)}/Q_{(k)} \geq \gamma$, где $\gamma \in (0,1)$ – некоторое пороговое значение, близкое к единице. Выполнение последнего условия означает, что при появлении нового центра кластера дисперсия меняется незначительно.

4.4. Алгоритм FOREL

Этот алгоритм был предложен Н.Г. Загоруйко и В.Н. Ёлкиной в 1967 году. В алгоритме FOREL (FORmal ELement), так же как и в алгоритме k -means, вычисляются центры тяжести кластеров. Но, в отличие от k -means, в качестве кластера рассматриваются не все ближайшие к данному центру элементы, а все элементы, находящиеся внутри сферы заданного радиуса r с центром в данной точке. Для фиксированного значения $r > 0$ и некоторого элемента $\mathbf{x} = \mathbf{e}^{(1)}$ обучающей выборки вычисляется формальный элемент $\mathbf{e}^{(2)}$ – центр тяжести всех векторов обучающей выборки Ξ , находящихся внутри круга $B_r(\mathbf{e}^{(1)})$ с центром в точке \mathbf{x}_1 и радиусом r . Затем вычисляется центр тяжести $\mathbf{e}^{(3)}$ всех элементов множества $B_r(\mathbf{e}^{(2)}) \cap \Xi$ и т.д. Можно показать (докажите!), что таким образом построенная последовательность формаль-

ных элементов $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \dots$ является сходящейся. Предел \mathbf{e} этой последовательности объявляется центром первого кластера. Далее из обучающей выборки Ξ удаляются все элементы, находящиеся внутри сферы $B_r(\mathbf{e})$ и аналогично находится центр следующего кластера и т.д.

Алгоритм FOREL

1. Выбирается некоторый элемент $\mathbf{x} \in \Xi$ обучающей выборки, $\mathbf{e}^{(1)} := \mathbf{x}$.
2. Вычисляется центр тяжести $\mathbf{e}^{(2)} = \frac{1}{|B_r(\mathbf{e}^{(1)}) \cap \Xi|} \sum_{\mathbf{x} \in B_r(\mathbf{e}^{(1)}) \cap \Xi} \mathbf{x}$. Выполнение пункта 2 повторяется до тех пор, пока последовательность $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \dots$ не стабилизируется в точке \mathbf{e} .
3. Полагаем $\Xi := \Xi \setminus B_r(\mathbf{e})$ и переходим к пункту 1. Условием останова алгоритма является $\Xi = \emptyset$.

Результаты работы алгоритма FOREL для трех разных значений r показаны на рис. 4.3. Здесь незакрашенные кружочки – элементы обучающей выборки, черные точки – формальные элементы.

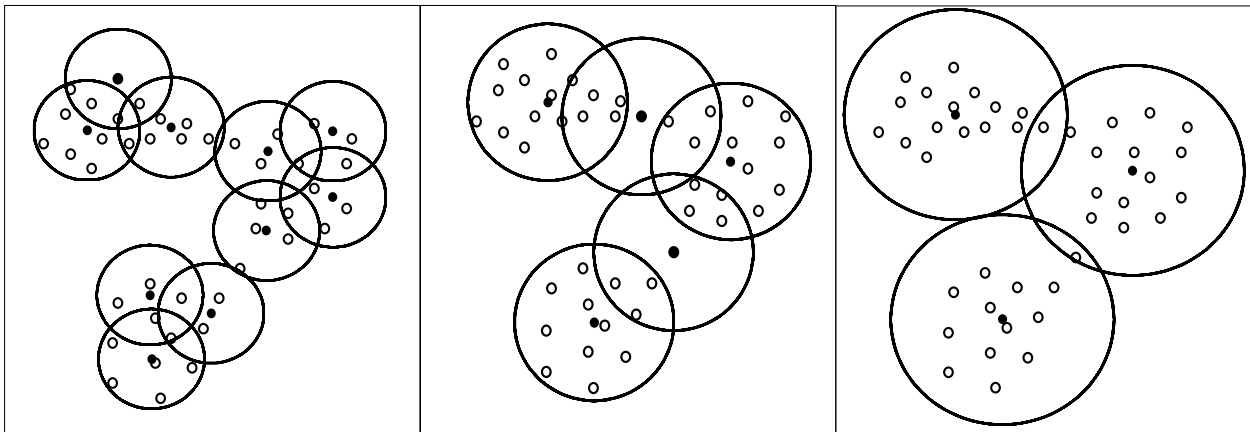


Рис. 4.3

В Приложении 1 приведен расчет кластеризации данных алгоритмом FOREL, выполненный с помощью пакета MathCad.

Результат работы алгоритма FOREL может существенно зависеть от выбора начальных точек – формальных элементов. Единственный параметр алгоритма – величина r подбирается исходя из задач кластеризации: если необходимо получить большие кластеры, то r следует увеличить, если же нужно описать структуру самих кластеров, то r следует уменьшить и кластеризовать «мелкие» кластеры. В этом случае алгоритм FOREL можно рассматривать как алгоритм предварительной кластеризации. Различные модификации алгоритма FOREL и его применение подробно рассмотрены в [11].

4.5. Алгоритм ИСОМАД (ISODATA)

Алгоритм ИСОМАД – Итеративный СамоОрганизирующийся Метод Анализа Данных (ISODATA – Iterative Self-Organizing Data Analysis Techniques) был разработан в 1965 году Бэллом (Ball G.) и Хэллом (Hall D.). Этот алгоритм является разновидностью алгоритма k -внутригрупповых средних и от-

личается от него введением некоторых эвристических процедур. С помощью таких процедур можно объединять два кластера в один, разделять один кластер на два и т.д.

Рассмотрим основные процедуры изменения числа кластеров.

1. Удаление кластеров. Если кластер содержит мало элементов $|X_i| < q_1$ (q_1 – параметр алгоритма ISODATA), то он удаляется, т.е. его элементы распределяются по другим кластерам, а центр кластера \mathbf{c}_i удаляется из списка центров кластеров.

2. Разделение кластеров. Если разброс элементов от центра кластера достаточно большой, или, другими словами, если дисперсия i -го кластера $D_i > q_2$, то i -й кластер разделяется на два кластера. Для разделения кластера вычисляются покомпонентные дисперсии:

$$D_{ik} = \frac{1}{|X_i|} \sum_{x_{jk} \in X_i} \|x_{jk} - c_{ik}\|^2, \quad k = 1, \dots, n.$$

Далее выбирается та l -я компонента, для которой $D_{il} > D_{is}$ для всех $s \neq l$, и осуществляется разделение i -го кластера по l -й компоненте. При этом пересчитываются новые центры кластеров \mathbf{c}' и \mathbf{c}'' .

Другой, более точный, способ деления кластеров состоит в вычислении «направления» в пространстве R^n , вдоль которого дисперсия кластера максимальна. Далее кластер разделяется на два гиперплоскостью, проходящей через центр кластера и перпендикулярной вычисленному направлению.

3. Слияние кластеров. Если расстояние между двумя какими-то центрами кластеров достаточно мало, то эти кластеры следует объединить в один кластер. Для реализации этой процедуры вычисляется расстояние между двумя центрами кластеров:

$$l_{ij} = \|\mathbf{c}_i - \mathbf{c}_j\| \quad \text{для всех } i \neq j.$$

Если окажется, что $l_{ij} < q_3$, то кластеры X_i и X_j следует объединить. Новый

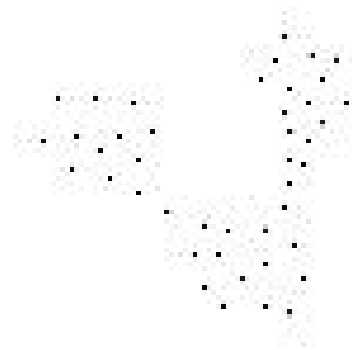
центр кластера вычисляется по формуле $\mathbf{c} = \frac{\mathbf{c}_i |X_i| + \mathbf{c}_j |X_j|}{|X_i| + |X_j|}$.

Алгоритм ISODATA может содержать и другие процедуры, регулирующие число кластеров.

Приложение 1. Кластеризация данных алгоритмом FOREL

1. Считывание точечных данных из файла в матрицу A и формирование массива точек (X,Y)

$\underline{A} := \text{READBMP}("D:/p3")$



A

A =

	0	1	2	3
0	255	255	255	255
1	255	255	255	255
2	255	255	255	255
3	255	255	255	255
4	255	255	255	255
5	255	255	255	...

```

points(A) :=
    k ← 0
    for i ∈ 0..rows(A) - 1
        for j ∈ 0..cols(A) - 1
            if Ai,j = 0
                | Xk ← i
                | Yk ← j
                | k ← k + 1
    for k ∈ 0..length(X) - 2
        if (Xk+1 - Xk)2 + (Yk - Yk+1)2 < 3
            | Xk ← -1
            | Yk ← -1
    s ← 0
    for k ∈ 0..length(X) - 1
        if Xk ≠ -1
            | X1s ← Xk
            | Y1s ← Yk
            | s ← s + 1
    ( X1 )
    ( Y1 )

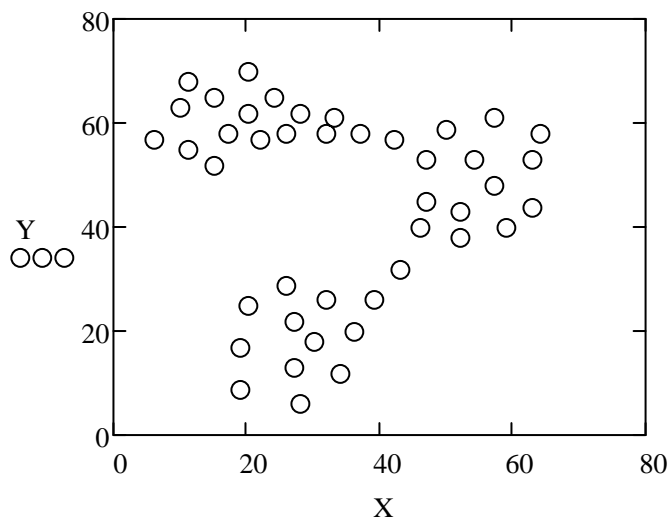
```

функция формирования из матрицы изображения точечных данных A массива точек (X,Y)

$$X := \text{points}(A)_0 \quad X^T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 6 & 10 & 11 & 11 & 15 & 15 & 17 & 19 & 19 & \dots \\ \hline \end{array}$$

$$Y := \text{points}(A)_1 \quad Y^T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 57 & 63 & 55 & 68 & 52 & 65 & 58 & 9 & 17 & \dots \\ \hline \end{array}$$

2. Функции алгоритма FOREL



$$d(C, S) := (C_0 - S_0)^2 + (C_1 - S_1)^2$$

$$\text{cent}(F, X, Y, r) := \left| \begin{array}{l} C1 \leftarrow (0 \ 0)^T \\ n \leftarrow 0 \\ \text{for } i \in 0.. \text{length}(X) - 1 \\ \quad \left| \begin{array}{l} C_i \leftarrow (X_i \ Y_i)^T \\ \text{if } d(C_i, F) < r^2 \\ \quad \left| \begin{array}{l} n \leftarrow n + 1 \\ C1 \leftarrow C1 + C_i \end{array} \right. \end{array} \right. \\ \frac{C1}{n} \end{array} \right.$$

функция определения центра масс точек массива (X,Y), содержащихся внутри круга с центром в точке F и радиусом r

$\text{ext}(F, X, Y, r) :=$	for $i \in 0.. \text{length}(X) - 1$ <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 10px;"> $C_i \leftarrow (X_i \ Y_i)^T$ if $d(C_i, F) > r^2$ <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 10px;"> $C1 \leftarrow C_i$ break </div> </div> for $i \in 0.. \text{length}(X) - 1$ <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 10px;"> $C_i \leftarrow (X_i \ Y_i)^T$ $C1 \leftarrow \text{augmen}(C1, C_i) \text{ if } d(C_i, F) > r^2 \wedge C_i \neq C1$ </div> $C1$	функция удаления из массива (X, Y) точек, содержащихся внутри круга с центром в точке F и радиусом r
-----------------------------	--	--

$\text{fore}(X, Y, r) :=$	$F0 \leftarrow (X_0 \ Y_0)^T$ $F \leftarrow F0$ $k \leftarrow \text{length}(X)$ while $k > 0$ <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 10px;"> $F1 \leftarrow \text{cent}(F0, X, Y, r)$ while $d(F0, F1) > 0.1$ <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 10px;"> $F0 \leftarrow F1$ $F1 \leftarrow \text{cent}(F1, X, Y, r)$ </div> $F \leftarrow \text{augmen}(F, F1)$ $X1 \leftarrow \left(\text{ext}(F1, X, Y, r)^T \right)^{\langle 0 \rangle} \text{ if } \text{ext}(F1, X, Y, r) \neq 0$ break otherwise $Y1 \leftarrow \left(\text{ext}(F1, X, Y, r)^T \right)^{\langle 1 \rangle} \text{ if } \text{ext}(F1, X, Y, r) \neq 0$ $X \leftarrow X1$ $Y \leftarrow Y1$ $F0 \leftarrow (X_0 \ Y_0)^T$ $k \leftarrow \text{length}(X)$ </div> $F \leftarrow \text{submatrix}(F, 0, 1, 1, \text{cols}(F) - 1)$ F	функция вычисления формальных элементов массива точек (X, Y) алгоритмом FOREL с параметром r
---------------------------	---	--

3. Результаты работы алгоритма FOREL

$r := 10$

$\underline{F} := \text{forel}(X, Y, r)$

$$F =$$

	0	1	2	3	4
0	14.111	26.286	26	30	20
1	59.667	13.857	26.667	60.333	...

$CX := (F^T)^{\langle 0 \rangle}$

$CY := (F^T)^{\langle 1 \rangle}$

$x(t, i) := F_{0,i} + r \cdot \cos(t)$

$y(t, i) := F_{1,i} + r \cdot \sin(t)$

